4pts

METHOD FOR CODING AND DECODING THE WIDENESS OF A SOUND SOURCE IN AN AUDIO SCENE

The invention relates to a method and to an apparatus for coding and decoding a presentation description of audio signals, especially for describing the presentation of sound sources encoded as audio objects according to the MPEG-4 Audio standard.

Background

MPEG-4 as defined in the MPEG-4 Audio standard ISO/IEC 14496-3:2001 and the MPEG-4 Systems standard 14496-1:2001 facilitates a wide variety of applications by supporting the representation of audio objects. For the combination of the audio objects additional information - the so-called scene description - determines the placement in space and time and is transmitted together with the coded audio objects.

For playback the audio objects are decoded separately and composed using the scene description in order to prepare a single soundtrack, which is then played to the listener.

For efficiency, the MPEG-4 Systems standard ISO/IEC 14496--1:2001 defines a way to encode the scene description in a binary representation, the so-called Binary Format for Scene Description (BIFS). Correspondingly, audio scenes are described using so-called AudioBIFS.

A scene description is structured hierarchically and can be represented as a graph, wherein leaf-nodes of the graph form the separate objects and the other nodes describes the processing, e.g. positioning, scaling, effects etc.. The appearance and behavior of the separate objects can be controlled using parameters within the scene description nodes.

Invention

The invention is based on the recognition of the following fact. The above mentioned version of the MPEG-4 Audio standard cannot describe sound sources that have a certain dimension, like a choir, orchestra, sea or rain but only a point source, e.g. a flying insect, or a single instrument. However, according to listening tests wideness of sound sources is clearly audible.

Therefore, a problem to be solved by the invention is to overcome the above mentioned drawback. This problem is solved by the coding method disclosed in claim 1 and the corresponding decoding method disclosed in claim 8.

In principle, the inventive coding method comprises the generation of a parametric description of a sound source which is linked with the audio signals of the sound source, wherein describing the wideness of a non-point sound source is described by means of the parametric description and a presentation of the non-point sound source is defined by multiple decorrelated point sound sources.

The inventive decoding method comprises, in principle, the reception of an audio signal corresponding to a sound source linked with a parametric description of the sound source. The parametric description of the sound source is evaluated for determining the wideness of a non-point sound source and multiple decorrelated point sound sources are assigned at different positions to the non-point sound source.

This allows the description of the wideness of sound sources that have a certain dimension in a simple and backwards compatible way. Especially, the playback of sound sources with a wide sound perception is possible with a monophonic signal, thus resulting in a low bit rate of the audio signal to be transmitted. An application is for example the monophonic transmission of an orchestra, which is not coupled to a fixed loudspeaker layout and allows to position it at a desired location.

Advantageous additional embodiments of the invention are disclosed in the respective dependent claims.

Drawings

Exemplary embodiments of the invention are described with reference to the accompanying drawings, which show in

Fig. 1 the general functionality of a node for describing the wideness of a sound source;

Fig. 2 an audio scene for a line sound source;

Fig. 3 an example to control the width of a sound source with an opening-angle relative to the listener;

Fig. 4 an exemplary scene with a combination of shapes to represent a more complex audio source.

Exemplary embodiments

Figure 1 shows an illustration of the general functionality of a node ND for describing the wideness of a sound source, in the following also named AudioSpatialDiffuseness node or AudioDiffusenes node.

This AudioSpatialDiffuseness node ND receives an audio signal AI consisting of one or more channels and will produce after decorrelation DECan audio signal AO having the same number of channels as output. In MPEG-4 terms this audio input corresponds to a so-called child, which is defined as a branch that is connected to an upper level branch and can be inserted in each branch of an audio subtree without changing any other node.

A diffuseSelection field DIS allows to control the selection of diffuseness algorithms. Therefore, in case of several AudioSpatialDiffuseness nodes each node can apply a different diffuseness algorithms, thus producing different outputs and ensuring a decorrelation of the respective outputs. A diffuseness node can virtually produce N different signals, but pass through only one real signal to the output of the node, selected by the diffuseSelect field. However, it is also possible that multiple real signals are produced by a signal diffuseness node and are put at the output of the node. Other fields like a field indicating the decorrelation strength DES could be added to the node, if required. This decorrelation strength could be measured e.g. with a cross-correlation function.

Table 1 shows possible semantics of the proposed AudioSpatialDiffuseness node. Children can be added or deleted to the node with the help of the addChildren field or removeChildren field, respectively. The children field contains the IDs, i.e. references, of the connected children. The diffuseSelect field and decorreStrength field are defined as scalar 32 bit integer values. The numChan field defines the number of channels at the output of the node. The phaseGroup field describes whether the output signals of the node are grouped together as phase related or not.

```
AudioSpatialDiffuseness  {
    eventin         MFNode  addChildren
    eventin         MFNode  removeChildren
    exposedField    MFNode  children            [ ]
    exposedField    SFInt32 diffuseSelect        1
    exposedField    SFInt32 decorreStrength      1
    field           SFInt32 numChan              1
    field           MFInt32 phaseGroup          [ ]
}
```

Table 1: Possible semantics of the proposed AudioSpatialDiffuseness Node

However, this is only one embodiment of the proposed node, different and/or additional fields are possible.

In the case of numChan greater than one, i.e. multichannel audio signals, each channel should be diffused separately.

For presentation of a non-point sound source by multiple decorrelated point sound sources the number and positions of the decorrelated multiple point sound sources have to be defined. This can be done either automatically or manually and by either explicit position parameters for an exact number of point sources or by relative parameters like the density of the point sound sources within a given shape. Furthermore, the presentation can be manipulated by using the intensity or direction of each point source as well as using the AudioDelay and AudioEffects nodes as defined in ISO/IEC 14496-1.

Figure 2 depicts an example of an audio scene for a Line Sound Source LSS. Three point sound sources S1, S2 and S3 are defined for representing the Line Sound Source LSS, wherein the respective position is given in cartesian coordinates. Sound source S1 is located at -3,0,0, sound source S2 at 0,0,0 and sound source S3 at 3,0,0. For the decorrelation of the sound sources different diffuseness algorithms are selected in the respective AudioSpatialDiffuseness Node ND1, ND2 or ND3, symbolized by DS=1,2 or 3.

Table 2 shows possible semantics for this example. A grouping with 3 sound objects POS1, POS2, and POS3 is defined. The normalized intensity is 0.9 for POS1 and 0.8 for POS2 and POS3. Their position is addressed by using the 'location'-field which in this case is a 3D- vector. POS1 is localized at the origin 0,0,0 and POS2 and POS3 are positioned -3 and 3 units in x direction relative to the origin, respectively. The 'spatialize'-field of the nodes is set to 'true', signaling that the sound has to be spatialized depending on the parameter in the 'location'-field. A 1-channel audio signal is used as indicated by numChan 1 and different diffuseness algorithms are selected in the respective AudioSpatialDiffuseness Node, as indicated by diffuseSelect 1,2 or 3. In the first AudioSpatialDiffuseness Node the AudioSource BEACH is defined, which is a 1-channel audio signal, and can be found at url 100. The second and third first AudioSpatialDiffuseness Node make use of the same AudioSource BEACH. This allows to reduce the computational power in an MPEG-4 player since the audio decoder converting the encoded audio data into PCM output signals only has to do the encoding once. For this purpose the renderer of the MPEG-4 player passes the scene tree to identify identical AudioSources.

```
# Example of a line sound source replaced by three point
sources
# using one single decoder output.

Group {
    children [
        DEF POS1 Sound {
            intensity 0.9
            location 0 0 0
            spatialize TRUE
            source AudioSpatialDiffuseness  {
                numChan 1
                diffuseSelect  1
                children [
                    DEF BEACH AudioSource {
                        numChan 1
                        url 100
                    }
                ]
            }

        DEF POS2 Sound {
            intensity 0.8
            location -3 0 0
            spatialize TRUE
            source AudioSpatialDiffuseness  {
                numChan 1
                diffuseSelect  2
                children [ USE BEACH]
            }

        DEF POS3 Sound {
            intensity 0.8
            location  3 0 0
            spatialize TRUE
            source AudioSpatialDiffuseness  {
                numChan 1
                diffuseSelect  3
                children [ USE BEACH]
            }
     ]
}
```

Table 2: Example of a Line Sound Source replaced by three Point Sources using one single Audio-Source.

According to a further embodiment primitive shapes are defined within the AudioSpatialDiffuseness nodes. An advantageous selection of shapes comprises e.g. a box, a sphere and a cylinder. All of these nodes could have a location field, a size and a rotation, as shown in table 3.

```
SoundBox / SoundSphere / SoundCylinder {
    eventin       MFNode   addChildren
    eventin       MFNode   removeChildren
    exposedField  MFNode   children                  [ ]
    exposedField  MFFloat  intensity           1.0
    exposedField  SFVec3f  location                  0,0,0
    exposedField  SFVec3f  size                      2,2,2
    exposedField  SFVec3f  rotationaxis              0,0,1
    exposedField  MFFloat  rotationangle             0.0
}
```

Table 3

If one vector element of the size field is set to zero a volume will be flat, resulting in a wall or a disk. If two vector elements are zero a line results.

Another approach to describe a size or a shape in a 3D coordinate system is to control the width of the sound with an opening-angle relative to the listener. The angle has a vertical and a horizontal component, 'widthHorizontal' and 'widthVertical', ranging from 0...2π with the location as its center. The definition of the widthHorizontal component φ is generally shown in Fig. 3. A sound source is positioned at location L. To achieve a good effect the location should be enclosed with at least two loudspeakers L1, L2. The coordinate system and the listeners location are assumed as a typical configuration used for stereo or 5.1 playback systems, wherein the listener's position should be in the so-called sweet spot given by the loudspeaker arrangement. The widthVertical is similar to this with a 90-degree x-y-rotated relation.

Furthermore, the above-mentioned primitive shapes can be combined to do more complex shapes. Fig. 4 shows a scene with two audio sources, a choir located in front of a listener L and audience to the left, right and back of the listener making applause. The choir consists out of one ***SoundSphere*** C and the audience consists out of three ***SoundBoxes*** A1, A2, and A3 connected with ***AudioDiffuseness*** nodes.

A BIFS example for the scene of figure 4 looks as shown in table 4. An audio source for the SoundSphere representing the Choir is positioned as defined in the location field with a size and intensity also given in the respective fields. A children field *APPLAUSE* is defined as an audio source for the first SoundBox and is reused as audio source for the second and third SoundBox. Furthermore, in this case the diffuseSelect field signals for the respective SoundBox which of the signals is passed through to the output.

```
## The Choir SoundSphere

    SoundSphere {
        location 0.0 0.0 -7.0   # 7 meter to the back
        size 3.0 0.6 1.5        # wide 3; height 0.6; depth 1.5
        intensity 0.9
        spatialize TRUE
        children [ AudioSource {
            numChan 1
            url 1
        }]
    }
## The audience consists out of 3 SoundBoxes

    SoundBox {                  # SoundBox to the left
        location -3.5 0.0 2.0   # 3.5 meter to the left
        size 2.0 0.5 6.0        # wide 2; height 0.5; depth 6.0
        intensity 0.9
        spatialize TRUE
        source AudioDiffusenes{
            diffuseSelect 1
            decorrStrength 1.0
            children [ DEF APPLAUSE AudioSource {
                numChan 1
                url 2
            }]
        }
    }
    SoundBox {                  # SoundBox to the rigth
        location 3.5 0.0 2.0    # 3.5 meter to the right
        size 2.0 0.5 6.0        # wide 2; height 0.5; depth 6.0
        intensity 0.9
        spatialize TRUE
        source AudioDiffusenes{
          diffuseSelect 2
          decorrStrength 1.0
          children [ USE APPLAUSE ]
      }
  }
  SoundBox {                      # SoundBox in the middle
      location 0.0 0.0 0.0        # 3.5 meter to the right
      size 5.0 0.5 2.0            # wide 2; height 0.5; depth 6.0
      direction 0.0 0.0 0.0 1.0     # default
      intensity 0.9
      spatialize TRUE
      source AudioDiffusenes{
          diffuseSelect 3
          decorrStrength 1.0
          children [ USE APPLAUSE ]
      }
  }
```

Table 4

In the case of a 2D scene it is still assumed that the sound will be 3D. Therefore it is proposed to use a second set of SoundVolume nodes, where the z-axis is replaced by a single float field with the name 'depth' as shown in table 5.

```
SoundBox2D / SoundSphere2D / SoundCylinder2D {
    eventin     MFNode  addChildren
    eventin     MFNode  removeChildren
    exposedField    MFNode  children                [ ]
    exposedField    MFFloat intensity               1.0
    exposedField    SFVec2f location                0,0
    exposedField    SFFloat locationdepth           0
    exposedField    SFVec2f size                    2,2
    exposedField    SFFloat sizedepth               0
    exposedField    SFVec2f rotationaxis            0,0
    exposedField    SFFloat rotationaxisdepth       1
    exposedField    MFFloat rotationangle           0.0
}
```

Table 5